UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/830,042	04/23/2004	John M. Holt	61130-8110.US01	9699

22918 7590 08/08/2008

PERKINS COIE LLP
P.O. BOX 1208
SEATTLE, WA 98111-1208

EXAMINER	
RUTTEN, JAMES D	
ART UNIT	PAPER NUMBER
2192	
MAIL DATE	DELIVERY MODE
08/08/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/830,042	HOLT, JOHN M.	
	Examiner	Art Unit	
	JAMES RUTTEN	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 April 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *7-9,14-18 and 24-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *7-9, 14-18, and 24-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *4/29/08 - 1 page only*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to Applicant's submission filed 4/29/08, responding to the 12/31/07 Office action which detailed the rejection of claims 7-9, 14-18, and 24-28. Claims 7, 14, 25, 27, and 28 have been amended, claims 1-6, 10-13, and 19-23 have been canceled, and claims 29 and 30 have been added. Claims 7-9, 14-18, and 24-30 remain pending in the application and have been fully considered and examined.

Response to Amendment/Arguments

2. The amendments to claims 14 and 28 have overcome the previous objections to those claims, which have likewise been withdrawn.

3. On page 8 filed 4/29/08, Applicants essentially argue that new claim elements directed to "without forming a distributed shared memory arrangement" are supported "at least in the description of Fig. 3 and the first paragraph of page 6 where it is explained that if each of the machines has a shared memory capability of 10 MB, then the total shared memory available to the application is not, as one might expect 10n MB (on the basis of Fig. 3) but rather only 10MB." However, it is not understood how the cited passages could support the new claim limitation. The passages actually appear to be describing a distributed shared memory arrangement.

Figure 3 is shown below:

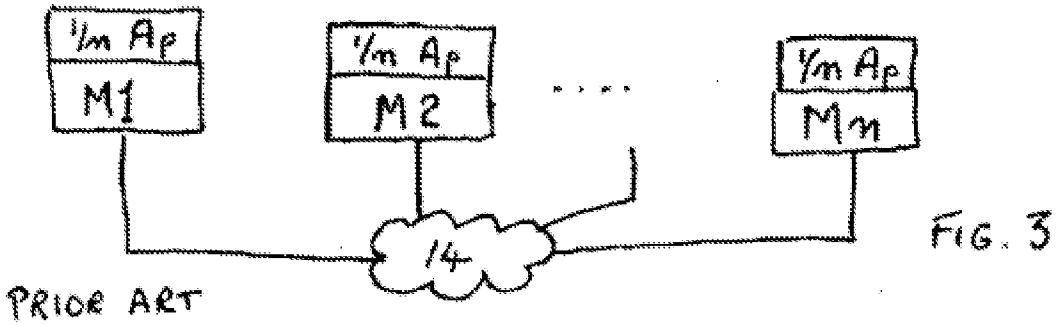

FIG. 3

PRIOR ART

Applicant appears to reference Fig. 3 to show an example of a distributed shared memory arrangement. However, the arguments seem to fall short of clearly presenting this argument as such. Nonetheless, it is noted that the description of Fig. 3 (found on page 2 lines 11-23) uses the term "distributed computing," and describes the process of partitioning a program into *m* tasks and distributing the tasks among *m* machines.

The first paragraph of page 6 describes Fig. 5, which appears below:

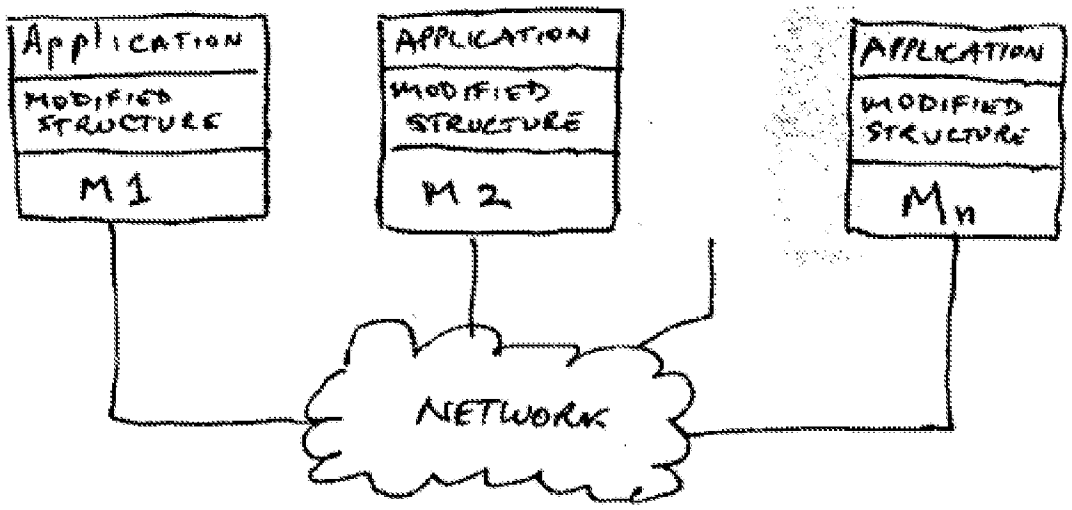

FIG. 5

As described in the last paragraph on page 5, each of the machines M1, M2, … Mn operates with the same code and data on each machine. As further described and cited by the Applicant at the

top of page 6, "the total shared memory available to the application is … 10MB." Note that the specification describes this arrangement in terms of being shared memory. Also note that the memory is distributed among *n* machines.

The passage describing Fig. 5 is presumed to contrast with the passage describing Fig. 3 in that the system of Fig. 5 does not partition the application for distribution as does the system of Fig. 3. However, Applicants have provided no description or support for the suggestion that Fig. 5 does not show a distributed shared memory arrangement. No description was found in the originally filed specification to support this suggestion. To the contrary, the specification clearly describes Fig. 5 in terms of being shared memory, and further describes a form of distribution (i.e. "notifies all other DRTs via the network of the changed value" - see page 7 lines 15-16). As such, the invention itself appears to qualify as a form of distributed shared memory.

Finally, no portion of the specification was found which clearly describes the new negative limitation "without forming a distributed shared memory arrangement." Applicants correctly point out on page 8 filed 4/29/08, that there is no prohibition against so-called negative limitations. However, Applicants are still bound by 35 U.S.C. § 112, first paragraph, to provide a full, clear, concise, and exact written description of the invention. Note that a lack of description is not the same as disclosure of a negative limitation. No portion of the originally filed disclosure could be found which fully, clearly, concisely, and exactly describes "without forming a distributed shared memory arrangement."

For the above reasons, Applicants' arguments regarding the new negative limitation "without forming a distributed shared memory arrangement" are not persuasive, and a new rejection under 35 U.S.C. § 112, first paragraph, is made below.

At the top of page 9 filed 4/29/08, Applicants essentially argue that the distributed computing environment of prior art of record Morshed is in contrast to claims 7 and 14 which recite an application "written to operate only on a single computer." It is noted that Morshed was not relied upon to teach this limitation. Rather, Chen teaches this limitation as cited in the previous Office action. Therefore, Applicants' argument is moot.

Also on page 9, Applicants argue that Morshed teaches away from the subject matter of claims 7 and 14. However, Applicants do not address the Chen reference with respect to this argument. Chen is relied upon to teach an application "written to operate only on a single computer." While Morshed teaches distributed computation, nothing in the disclosure expressly discourages or prevents the combination with Chen. Therefore, Applicants argument is not persuasive.

On page 10 filed 4/29/08, Applicants address the Chen reference, and essentially argue that Chen teaches a distributed shared memory system, and that the limitations "without forming a distributed shared memory arrangement" and "a different independent local memory accessible only by a corresponding portion of the application program" are incompatible with a distributed shared memory system. However, as discussed above, Applicants' system appears to implement a distributed shared memory arrangement (e.g. Fig. 5), and this is not incompatible with having a "different independent local memory" which is disclosed by Morshed (i.e. implicitly through use of MS Windows ®). Further, any distributed memory in Chen is necessarily "independent" inasmuch as the distributed memory systems of Fig. 5 are independent.

Near the bottom of page 10, Applicants essentially argue that the applications of Chen are written to operate in parallel on a distributed memory system, and not "written to operate only on a single computer" as required by the claims. However, the fact that Chen discloses a distributed memory system does not preclude an application which runs on it from being written to operate only on a single computer. Chen makes this clear in the second column on page 1. Further, see the second column on page 2, i.e. "the objects need not be explicitly declared as shared." Thus, without explicitly sharing any variables, the application is "written" to operate on a single computer. While Chen's "MultiJav" system may itself be a distributed shared memory implementation, the applications which are developed for it are based simply upon the standard Java specifications which allow it to "be run on a standalone machine." If any application written for MultiJav can be run on a standalone machine, then the application must be written to operate only on a single computer, regardless of whether or not it can also be run in a distributed system.

On page 12, Applicants essentially argue that the Buhlman reference introduce delay paths which slow transmission speed, which is not in the spirit of Applicants' invention. However, these delay paths are used to synchronize data as it is distributed to remote machines through the network. This effect is understood to be similarly described on page 7 lines 14-16 of the originally filed specification:

> At this stage the processing of that thread is **halted**, and the same thread **notifies all other DRTs** via the network of the changed value.

Therefore, Applicants' argument is not persuasive.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 7-9, 14-18, and 24-30 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Independent claims 7, 14, and 28 recite "a plurality of computers interconnectable via a communications link without forming a distributed shared memory arrangement," and independent claim 27 recites "said different independent local memory within each said different computer not forming a distributed shared memory arrangement." These claim limitations are understood in the context of Applicants arguments on pages 8-9 filed 4/29/08, to mean that data stored in the memory of one computer is not shared by distributing the values to another computer in order to accomplish the execution of an application program. In the broadest sense, this is what a distributed shared memory arrangement does, and there are no other full, clear concise, or exact descriptions of such in the originally filed specification. Further, no full, clear concise, or exact descriptions of execution of an application through a plurality of computers interconnectable via a communications link *without forming a distributed shared memory arrangement* were found. Without the formation of a distributed shared memory, how could data from one machine be utilized in another machine which is executing the same application? Without data from the first machine, the second machine might operate on

old data and present an incorrect result. No portion of the specification was found to explain the distributed execution of an application without a distributed shared memory arrangement. For the purpose of further examination, the limitations "without forming a distributed shared memory arrangement" and "not forming a distributed shared memory arrangement" will be interpreted in light of Applicants' comments in reference to Fig. 3 on page 8 filed 4/29/08.

The remaining claims are dependent on one of the aforementioned independent claims, and are rejected for the same reasons.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 7, 8, 24, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,760,903 to Morshed et al. (hereinafter "Morshed") in view of "MultiJav: A Distributed Shared Memory System Based on Multiple Java Virtual Machines" by Chen et al. (hereinafter "Chen").

In regard to claim 7, Morshed discloses:

A method of loading an application program ...onto each of a plurality of computers, See column 20:60-61:

> Byte code may be instrumented by instrumenting each class as the class is loaded by the VM runtime system.

the plurality of computers being interconnectable via a communications link without forming a distributed shared memory arrangement, See Figure 29 and associated text in column 32:50-57, e.g. "network"; Note that Morshed does not partition and distribute the application throughout the system to form a *distributed shared memory arrangement* as described in Applicants' arguments on page 8 filed 4/29/08.

and different portions of said application program being simultaneously executable on each different one of the plurality of computers with each different one of the plurality of computers having a different independent local memory accessible only by a corresponding portion of the application program, See column 34 lines 39-42:

> In this example, the COM DLL may be used for facilitating interprocess communication, for example, as between a client and a server as well as between any two server systems.

Also see FIG. 29 and associated text in column 32 lines 50-57, e.g. elements 1000 and 1016a. This figure depicts separate computer systems which Morshed describes in terms of Intel Pentium processors running the MS Windows® operating systems which implicitly provides access to independent local memory by a corresponding portion of software.

the method comprising the steps of:

loading the application program onto each different computer of said plurality of computers; and See column 20:60-61, also column 34 lines 39-42 as cited above.

modifying the application program on each said different computer before execution of said corresponding portion of the application program on each said different computer. See column 20:60-61:

> Byte code may be instrumented by instrumenting each class as the class is loaded by the VM runtime system

Morshed does not expressly disclose: *written to operate only on a single computer*. However, Chen teaches loading an application written to operate only on a single computer, on different computers. See page 1 right column: "Thus, the same code can be run on a standalone machine without modification." It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Chen's teaching of simultaneous execution of uniprocessor programs with Morshed's distribution in order to provide distributed computing while maintaining portability and adherence to standard specifications as suggested by Chen (see page 1 right column).

In regard to claim 8, the above rejection of claim 7 is incorporated. Morshed further discloses: *wherein the step of modifying the application program is different for different computers.* See column 33:28-31, e.g. "different activities."

In regard to claim 24, the above rejection of claim 7 is incorporated. Morshed does not expressly disclose: *wherein said program written to operate on only a single computer is a program written to execute within a local processor or processors and local memory coupled to the processor or processors within the single computer.* However, Chen teaches this on page 1 right column. Chen's "standalone machine" execution necessarily executes within a local processor using local memory.

In regard to claim 27, Morshed discloses:

said different independent local memory within each said different computer not forming a distributed shared memory arrangement being accessible during execution of said application program and said different portions of said application program only by the different portion of the application program actually executing within the different computer, See Morshed column 34 lines 39-42:

> In this example, the COM DLL may be used for facilitating interprocess communication, for example, as between a client and a server as well as between any two server systems.

Note that clients and servers are distinct systems that only control their respective local resources. All further limitations have been addressed in the above rejection of claim 7.

8. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morshed and Chen in view of U.S. Patent No. 5,802,585 to Scales et al. (hereinafter "Scales").

In regard to claim 9, the above rejection of claim 7 or 8 is incorporated. Morshed and Chen do not expressly disclose further elements of claim 9. However, Scales teaches: *wherein said modifying step comprises:*

(i) detecting instructions which share memory records See Scales column 4:38-42, e.g. "coherency."

(ii) listing all such shared memory records and providing a naming tag for each listed memory record See column 6:20-21, e.g. "table," and column 11:6-10, e.g. "ID."

(iii) detecting those instructions which write to, or manipulate the contexts of, any of said listed memory records, and See column 4:30-32, e.g. "stores."

(iv) generating an alert instruction corresponding to each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record. See column 1:43-49, e.g. "message passing" and column 32-35, e.g. "miss check."

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Scales' shared memory with Morshed's loading in order to provide coherency (see Scales column 1:20-26).

9. Claims 14, 15, 26, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Scales in view of Chen.

In regard to claim 14, Scales discloses:

A method of compiling or modifying an application program ...to run simultaneously on each one of a plurality of computers interconnectable via a communications link See Figure 2 with supporting disclosure in column 4:29-30, e.g. "programs 215 are instrumented", Also see column 4 lines 18-19: "During operation of the system 200, instructions of the programs 215 are executed by the processors 211." This passage shows that a portion of the program runs on one of a plurality of computers.

with different portions of said application program being simultaneously executable on different ones of said plurality of computers with each one of the plurality of computers having an independent local memory accessible only by the corresponding

portion of the application program, See column 5 lines 6-8, e.g. "store private data only operated on by a local processor."

said method comprising the steps of:

(i) detecting instructions which share memory records See Scales column 4:38-42, e.g. "coherency."

(ii) listing all such shared memory records and providing a naming tag for each listed memory record See column 6:20-21, e.g. "table," and column 11:6-10, e.g. "ID."

(iii) detecting those instructions which write to, or manipulate the contents of, any of said listed memory records, and See column 4:30-32, e.g. "stores."

(iv) generating an alert instruction following each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record. See column 1:43-49, e.g. "message passing" and column 32-35, e.g. "miss check."

Scales does not expressly disclose: *without forming a distributed shared memory arrangement* or *written to operate only on a single computer.* However, Chen teaches a distributed system that does not conform to the discussion of a distributed shared memory system by partitioning an application as described in Applicants' arguments in connection with Fig. 3 on page 8 filed 4/29/08. Instead, spawned threads are distributed among the system (see Chen section 3 at the bottom of page 2). Chen also discloses loading an application written to operate only on a single computer, on different computers. See page 1 right column: "Thus, the same code can be run on a standalone machine without modification." It would have been obvious to one of ordinary skill in the art at the time

the invention was made to use Chen's teaching of simultaneous execution of uniprocessor programs with Scales' distribution in order to provide distributed computing while maintaining portability and adherence to standard specifications as suggested by Chen (see page 1 right column).

In regard to claim 15, the above rejection of claim 14 is incorporated. Scales further discloses: *carried out prior to loading the application program onto each said computer.* See column 4 lines 29-30, e.g. "prior to execution."

In regard to claim 26, the above rejection of claim 14 is incorporated. Scales does not expressly disclose: *wherein said program written to operate on only a single computer is a program written to execute within a local processor or processors and local memory coupled to the processor or processors within the single computer.* However, Chen teaches this on page 1 right column. Chen's "standalone machine" execution necessarily executes within a local processor using local memory.

In regard to claim 28, Scales does not expressly disclose: *said different independent local memory within each said different computer being accessible during execution of said application program and said different portions of said application program only by the different portion of the application program actually executing within the different computer,* However, Chen teaches this through a memory coherence model. See section 3.3 on page 5: "The memory of site *q* is required to be consistent

with site *p*, which requires all the updates to shared variables at site *p* to be visible at site *q*." It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Chen's memory coherence with Scales distributed execution in order to provide data coherency as suggested by Chen. All further limitations have been addressed in the above rejection of claim 14.

10. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Scales and Chen as applied to claim 14 above, and further in view of Morshed.

In regard to claim 16, the above rejection of claim 14 is incorporated. Scales and Chen do not expressly disclose: *carried out during loading of the application program onto each said computer.* However, Morshed teaches instrumenting during loading. See column 20:60-61. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Morshed's teaching of loading with Scales' modification in order to automatically instrument a class instance (see Morshed column 20:61-65).

11. Claims 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Scales and Chen as applied to claim 14 above, and further in view of U.S. Patent Application Publication 2004/0163077 by Dimpsey et al. (hereinafter "Dimpsey.")

In regard to claim 17, the above rejection of claim 14 is incorporated. Scales and Chen do not expressly disclose: *carried out by just-in-time compilation.* However, Dimpsey teaches just-in-time compilation. See paragraph [0050]. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dimpsey's compiler with Scales' modification in order to increase execution speed while reducing compilation time as inherently provided by just-in-time compilation.

In regard to claim 18, the above rejection of claim 14 is incorporated. Scales and Chen do not expressly disclose: *carried out by re-compilation after loading.* However, Dimpsey teaches dynamic instrumentation after loading code. See Abstract. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dimpsey's dynamic instrumentation in order to minimize system perturbation (see Dimpsey's Abstract). Note that page 9 lines 6-8 of Applicant's specification inform broad interpretation of the concept of "compilation."

12. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morshed and Chen as applied to claim 7 above, and further in view of U.S. Patent No. 6,862,608 to Buhlman et al. (hereinafter "Buhlman").

In regard to claim 25, the above rejection of claim 7 is incorporated. Morshed and Chen do not expressly disclose: *wherein each of the computers operates with the same application program and data and thus all of the plurality of computers have the*

same application program and data. However, Buhlman teaches that programs can be operated using the same program and data. See column 1 lines 30-38. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Buhlman's teaching of multiple copies of shared memory with Morshed's distributed execution in order to reduce latency as suggested by Buhlman.

In regard to claims 29 and 30, the above rejections of claims 7 and 25 are respectively incorporated. Morshed does not expressly disclose: *eliminate clock cycle delays that would otherwise be associated with one or said plurality of computers reading memory physically located in a different one or ones of the plurality of computers formed in a distributed shared memory arrangement.* However, Buhlman discloses this by way of providing updated data values to each of the distributed processors. Each processor has its own copy of data and therefore does not have to request and wait for a data item from another processor. See column 3 lines 48-62. Thus delays associated with waiting for data from another processor are eliminated. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Buhlman's teaching of multiple copies of shared memory with Morshed's distributed execution in order to reduce latency as suggested by Buhlman.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to JAMES RUTTEN whose telephone number is (571)272-3703. The examiner can normally be reached on M-F 9:00-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/J. R./
Patent Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192